

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Koichi KONDO

GAU:

SERIAL NO: NEW APPLICATION

EXAMINER:

FILED: HERewith

FOR: SIMULATION METHOD, APPARATUS, AND COMPUTER PROGRAM USING HYBRID MODEL

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS  
ALEXANDRIA, VIRGINIA 22313

SIR:

☐ Full benefit of the filing date of U.S. Application Serial Number , filed , is claimed pursuant to the provisions of 35 U.S.C. §120.

☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of 35 U.S.C. §119(e): Application No. Date Filed

☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
Japan	2002-344228	November 27, 2002

Certified copies of the corresponding Convention Application(s)

☒ are submitted herewith

☐ will be submitted prior to payment of the Final Fee

☐ were filed in prior application Serial No. filed

☐ were submitted to the International Bureau in PCT Application Number

Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.

☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and

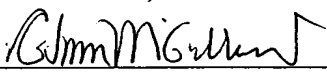
☐ (B) Application Serial No.(s)

☐ are submitted herewith

☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,  
MAIER & NEUSTADT, P.C.

  
\_\_\_\_\_  
Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland  
Registration Number 21,124

Customer Number

22850

Tel. (703) 413-3000  
Fax. (703) 413-2220  
(OSMMN 05/03)

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 2 年 1 1 月 2 7 日  
Date of Application:

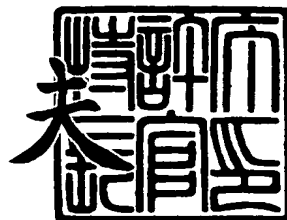
出 願 番 号            特 願 2 0 0 2 - 3 4 4 2 2 8  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 2 - 3 4 4 2 2 8 ]

出      願      人            株 式 会 社 東 芝  
Applicant(s):

2 0 0 3 年    7 月 1 8 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康



【書類名】 特許願

【整理番号】 A000204835

【提出日】 平成14年11月27日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/00

【発明の名称】 ハイブリッドモデルを用いるシミュレーション方法およびプログラム

【請求項の数】 10

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝研究開発センター内

【氏名】 近藤 浩一

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

## 【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

## 【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

## 【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

## 【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

## 【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ハイブリッドモデルを用いるシミュレーション方法およびプログラム

【特許請求の範囲】

【請求項 1】 ハイブリッドモデルを利用して、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーション方法であって、

前記ハイブリッドモデルの記述データを解析し、状態遷移に伴う連続系方程式の切り替えに関する記述と、前記ハイブリッドモデルの記述データ中に出現する連続系方程式の記述とを抽出する解析ステップと、

前記解析ステップにおいて抽出された状態遷移に伴う連続系方程式の切り替えに関する記述から、該連続系方程式の切り替えが必要となる条件及び対象となる連続系方程式の関係を表すテーブルを生成するテーブル生成ステップと、

前記解析ステップにおいて抽出された連続系方程式の記述から、該連続系方程式についての内部データ表現を生成する内部データ表現生成ステップと、

イベントの生起に応じて前記テーブルを参照し、前記条件に従って連続系方程式を有効なものに切り替える切り替えステップと、

前記切り替えステップにおいて切り替えられた連続系方程式に対応する前記内部データ表現を用い、前記時間軸に沿った数値積分により該連続系方程式を解いて前記機構の挙動を表すデータを出力するシミュレーション実行ステップと、を具備することを特徴とするシミュレーション方法。

【請求項 2】 前記切り替えステップは、少なくとも一つの前記連続系方程式について、該連続系方程式が有効であるか否かを表すフラグを操作することにより、該連続系方程式を有効なものに切り替えることを特徴とする請求項 1 に記載のシミュレーション方法。

【請求項 3】 前記イベントは、前記機構を制御する機構制御ソフトウェアを含む外部からの制御信号又は所定の内部変数の評価結果を含むことを特徴とする請求項 1 又は 2 に記載のシミュレーション方法。

【請求項 4】 前記シミュレーション実行ステップにおける数値積分の演算結果に基づいてキネマティックスシミュレーションを実行するステップをさらに

具備することを特徴とする請求項 1 乃至 3 のいずれかに記載のシミュレーション方法。

【請求項 5】 ハイブリッドモデルを利用して、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーション方法であって、

前記ハイブリッドモデルの記述データを解析し、状態遷移に伴う連続系方程式の切り替えに関する記述と、前記ハイブリッドモデルの記述データ中に出現する連続系方程式の記述とを抽出する解析ステップと、

前記解析ステップにおいて抽出された連続系方程式の記述に基づいて、第 1 のプログラムを生成する第 1 生成ステップと、

前記解析ステップにおいて抽出された状態遷移に伴う連続系方程式の切り替えに関する記述に基づいて、第 2 のプログラムを生成する第 2 生成ステップと、

前記第 1 生成ステップにおいて生成された第 1 のプログラムを実行することにより、前記連続系方程式についての内部データ表現を生成するステップと、

前記第 2 生成ステップにおいて生成された第 2 のプログラムを実行することにより、イベントの生起に応じて前記条件に従って連続系方程式を有効なものに切り替える切り替えステップと、

前記切り替えステップにおいて切り替えられた連続系方程式に対応する前記内部データ表現を用い、前記時間軸に沿った数値積分により該連続系方程式を解いて前記機構の挙動を表すデータを出力するシミュレーション実行ステップと、を具備することを特徴とするシミュレーション方法。

【請求項 6】 前記切り替えステップは、少なくとも一つの前記連続系方程式について、該連続系方程式が有効であるか否かを表すフラグを操作することにより、該連続系方程式を有効なものに切り替えることを特徴とする請求項 5 に記載のシミュレーション方法。

【請求項 7】 前記イベントは、前記機構を制御する機構制御ソフトウェアを含む外部からの制御信号又は所定の内部変数の評価結果を含むことを特徴とする請求項 5 又は 6 に記載のシミュレーション方法。

【請求項 8】 前記シミュレーション実行ステップにおける数値積分の演算結果に基づいてキネマティックスシミュレーションを実行するステップをさらに

具備することを特徴とする請求項5乃至7のいずれかに記載のシミュレーション方法。

【請求項9】 ハイブリッドモデルを利用して、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーションプログラムであって、

前記ハイブリッドモデルの記述データを解析し、状態遷移に伴う連続系方程式の切り替えに関する記述と、前記ハイブリッドモデルの記述データ中に出現する連続系方程式の記述とを抽出する解析手順と、

前記解析手順において抽出された状態遷移に伴う連続系方程式の切り替えに関する記述から、該連続系方程式の切り替えが必要となる条件及び対象となる連続系方程式の関係を表すテーブルを生成するテーブル生成手順と、

前記解析手順において抽出された連続系方程式の記述から、該連続系方程式についての内部データ表現を生成する内部データ表現生成手順と、

イベントの生起に応じて前記テーブルを参照し、前記条件に従って連続系方程式を有効なものに切り替える切り替え手順と、

前記切り替え手順において切り替えられた連続系方程式に対応する前記内部データ表現を用い、前記時間軸に沿った数値積分により該連続系方程式を解いて前記機構の挙動を表すデータを出力するシミュレーション実行手順と、  
をコンピュータに実行させるためのシミュレーションプログラム。

【請求項10】 ハイブリッドモデルを利用して、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーションプログラムであって、

前記ハイブリッドモデルの記述データを解析し、状態遷移に伴う連続系方程式の切り替えに関する記述と、前記ハイブリッドモデルの記述データ中に出現する連続系方程式の記述とを抽出する解析手順と、

前記解析手順において抽出された連続系方程式の記述に基づいて、第1のプログラムを生成する第1生成手順と、

前記解析手順において抽出された状態遷移に伴う連続系方程式の切り替えに関する記述に基づいて、第2のプログラムを生成する第2生成手順と、

前記第1生成手順において生成された第1のプログラムを実行することにより、前記連続系方程式についての内部データ表現を生成する手順と、

前記第2生成手順において生成された第2のプログラムを実行することにより、イベントの生起に応じて前記条件に従って連続系方程式を有効なものに切り替える切り替え手順と、

前記切り替え手順において切り替えられた連続系方程式に対応する前記内部データ表現を用い、前記時間軸に沿った数値積分により該連続系方程式を解いて前記機構の挙動を表すデータを出力するシミュレーション実行手順と、  
をコンピュータに実行させるためのシミュレーションプログラム。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

本発明は、コンピュータを用いて機械等の挙動のシミュレーションを行うための方法およびプログラムに関し、特に本発明はハイブリッドモデルを用いるものである。

##### 【0002】

#### 【従来の技術】

現在、コンピュータを用いて機械やプラント等の挙動のシミュレーションを行う際に、ハイブリッドモデリングと呼ばれる手法が使われている。ハイブリッドモデルを用いたシミュレーションは「ハイブリッドシミュレーション」と呼ばれている。このようなシミュレーション挙動をするシステムを「ハイブリッドシステム」と呼ぶこともある。

##### 【0003】

シミュレーションの目的で作成されるハイブリッドモデルは、概念的には常微分方程式や代数方程式を連立させた連立方程式によって表現される連続系モデルと、イベント発生に伴う状態遷移を表現するための状態遷移モデルとを組み合わせたモデルである。ハイブリッドモデルによれば、連続系モデルで表現される状態が外部からのイベントなどにより瞬時に切り替わるシステムを表現することができる。

##### 【0004】

ハイブリッドモデルを記述するための言語として、米国ゼロックス社（商標）



のパロアルト研究所にて創作されたHCC (Hybrid Concurrent Constraint Programming)と呼ばれる言語がある(下記特許文献1および非特許文献1参照。)。HCCは発展途上にあり、現在も米国NASAのエームズ研究所において研究が進められている。HCCは制約処理プログラミング(コンストレイントプログラミング)と呼ばれる技術の一種であり、連続系モデルを表現する常微分方程式や代数方程式を制約として扱い、これら方程式をそのまま順不同で記述することができる。このような制約記述に、状態遷移を制御する記述を付加してHCC言語のハイブリッドモデルは完成される。HCCによれば、方程式をそのまま制約として羅列する(プログラミングする)ことができ、複雑なモデルを記述可能である。

#### 【0005】

このようにハイブリッドモデルの技術を用いれば、系の特性を常微分方程式などでモデル表現し、初期状態から時間の推移に従ってどのような挙動を示すかをシミュレーションすることができる。

#### 【0006】

微分方程式などで表現可能な対象や現象を的確にモデル化できるハイブリッドモデルの技術の応用例として、ソフトウェアにより機構が制御されるメカトロニクス機器の機構シミュレーションがある。かかる機構シミュレーションによれば、機構の実機が存在しない状況においても、当該機構を制御する制御ソフトウェアのプロトタイピング、テスト、あるいはデバッグなどが行えるようになる。

#### 【0007】

しかし、ハイブリッドモデルを扱うことのできる公知のプログラム言語は、必ずしもメカトロニクス機器の機構シミュレーションに応用することを目的に開発されたものではなく、このため次のような問題点がある。

#### 【0008】

例えば米国ゼロックス社(商標)のHCCは、インタプリタ型のプログラミング言語であり、実行中にGC (Garbage Collection:ごみ集め)と呼ばれる処理が起動される。この処理が完了するまで、シミュレーション(のプログラム)は一時的に停止させられてしまう。これによりシミュレーションの実行時間を正確

に把握できない。アプリケーションの実装に不向きな点は他にもある。例えばシミュレーションのプログラム作成が極めて難易度が高い点である。たとえば制御ソフトウェアが機構に送信するアクチュエータへの動作コマンドなどを、シミュレータの外部から制御信号として受け取るような構成とする場合、外部関数などを個別に定義する必要がある上、プログラミングにおいてかなりの工夫を必要とする。

【0009】

【特許文献1】

米国特許第5,831,853号明細書

【0010】

【非特許文献1】

インターネット<URL: <http://www2.parc.com/spl/projects/mbc/publications.html#cclanguages>>

【0011】

【発明が解決しようとする課題】

本発明はかかる事情を考慮してなされたものであり、ハイブリッドモデルを用いて複雑な機構系を簡便かつ正確にモデル化でき、該機構系を制御する制御ソフトウェアとの連携シミュレーションにも好適なシミュレーション方法およびプログラムを提供することを目的とする。

【0012】

【課題を解決するための手段】

上記課題を解決し目的を達成するために本発明は次のように構成されている。

【0013】

本発明に係るシミュレーション方法は、ハイブリッドモデルを利用して、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーション方法であって、前記ハイブリッドモデルの記述データを解析し、状態遷移に伴う連続系方程式の切り替えに関する記述と、前記ハイブリッドモデルの記述データ中に出現する連続系方程式の記述とを抽出する解析ステップと、前記解析ステップにおいて抽出された状態遷移に伴う連続系方程式の切り替えに関する記述から、該

連続系方程式の切り替えが必要となる条件及び対象となる連続系方程式の関係を表すテーブルを生成するテーブル生成ステップと、前記解析ステップにおいて抽出された連続系方程式の記述から、該連続系方程式についての内部データ表現を生成する内部データ表現生成ステップと、イベントの生起に応じて前記テーブルを参照し、前記条件に従って連続系方程式を有効なものに切り替える切り替えステップと、前記切り替えステップにおいて切り替えられた連続系方程式に対応する前記内部データ表現を用い、前記時間軸に沿った数値積分により該連続系方程式を解いて前記機構の挙動を表すデータを出力するシミュレーション実行ステップと、を具備することを特徴とするシミュレーション方法である。

#### 【0014】

また、本発明に係るシミュレーション方法は、ハイブリッドモデルを利用して、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーション方法であって、前記ハイブリッドモデルの記述データを解析し、状態遷移に伴う連続系方程式の切り替えに関する記述と、前記ハイブリッドモデルの記述データ中に出現する連続系方程式の記述とを抽出する解析ステップと、前記解析ステップにおいて抽出された連続系方程式の記述に基づいて、第1のプログラムを生成する第1生成ステップと、前記解析ステップにおいて抽出された状態遷移に伴う連続系方程式の切り替えに関する記述に基づいて、第2のプログラムを生成する第2生成ステップと、前記第1生成ステップにおいて生成された第1のプログラムを実行することにより、前記連続系方程式についての内部データ表現を生成するステップと、前記第2生成ステップにおいて生成された第2のプログラムを実行することにより、イベントの生起に応じて前記条件に従って連続系方程式を有効なものに切り替える切り替えステップと、前記切り替えステップにおいて切り替えられた連続系方程式に対応する前記内部データ表現を用い、前記時間軸に沿った数値積分により該連続系方程式を解いて前記機構の挙動を表すデータを出力するシミュレーション実行ステップと、を具備することを特徴とするシミュレーション方法である。

#### 【0015】

#### 【発明の実施の形態】

以下、図面を参照しながら本発明の実施形態を説明する。

【0016】

(第1実施形態)

図1は、本発明の第1実施形態に係る機構シミュレータの概略構成を示すブロック図である。

【0017】

本発明の第1実施形態に係る機構シミュレータ（ハイブリッドモデルシミュレーション部101）はモデル方程式制御情報解析部111と、方程式構文解析部112と、ハイブリッドモデルシミュレーション実行部102とにより構成されている。ハイブリッドモデル記述104は、例えばHCCのようなハイブリッドモデル記述言語で記述されたソースプログラムであって、本実施形態に係る機構シミュレータ101への入力である。制御信号106も機構シミュレータ101への入力であり、この制御信号106は後述する機構制御ソフトウェアないしは機構制御ソフトウェアのシミュレータから与えられる。また、本実施形態に係る機構シミュレータ101からの出力は、シミュレーション結果としての変数値の演算結果およびその時間履歴であり、記憶部105に対して出力される。

【0018】

図1に示すように、ハイブリッドモデルシミュレーション実行部102は、モデル方程式制御情報記憶部113と、方程式データ記憶部114と、連続系方程式切り替え部115と、連続系シミュレーション部103とにより構成されている。なお、本実施形態は一般的なコンピュータを用いて構成することができ、その基本的なハードウェア構成として、図示しないが中央演算ユニット（CPU）、メモリ、外部記録装置、通信インターフェース（I/F）、および表示装置やキーボード、マウス等の入力装置を備える。また、これらのハードウェアを制御するためのオペレーティングシステム（OS）を備える。また、本発明の実施形態に係る機構シミュレータは、このようなオペレーティングシステム上で動作するアプリケーションソフトウェアとして実装することができる。

【0019】

このような本実施形態に係る機構シミュレータの構成及びその処理手順につい

て説明する前に、ここではまずハイブリッドモデル 104 がどのように記述されるかを具体例を挙げて説明する。

### 【0020】

図 2 および図 3 は、具体例に係るハイブリッドモデルの記述対象である機械装置を示す図であって、この機械装置は、バルブ 301、バネ 303、およびピストン 302 を備えた簡素な構造を有するシリンダ装置である。

### 【0021】

バルブ 301 は外部からの指令（イベント）に応じて開閉動作する。これによりシリンダ装置内の空気の流れを図 2 のように右側に変更するイベントを以下、「Right」と呼び、空気の流れを図 3 のように左側に変更するイベントを「Left」と呼ぶ。図 2 は、バルブ 301 に Right のイベントが与えられた状態を示しており、ピストン 302 には紙面左向きへの力が作用している。この状態を示す運動方程式はシリンダ装置の下部に示してあるように、「 $-F = m x''$ 」である。これに対し図 3 は、バルブ 301 に Left のイベントが与えられた状態を示しており、空気の流れの向きが変わり、運動方程式は同図のように「 $F = m x''$ 」に変化している。

### 【0022】

図 4 は、このような状態変化とそれぞれの状態に対応する運動方程式を状態遷移図として表現したものである。ハイブリッドモデルは、この図 4 に示されるような状態遷移と、各状態の記述が微分方程式や代数方程式で表現されるものを指す。図 4 によれば、状態が 2 つあって、かかる 2 つの状態間に状態遷移が存在することがわかる。

### 【0023】

図 5 は、図 4 の状態遷移図をもとに具体的なハイブリッドモデルの内容を HCC (Hybrid Concurrent Constraint Programming) 言語で記述したプログラムの一例を示す図である。図 5 において、（ソース）プログラムの論理行番号を仮に L1～L10 とする。L3、L4、および L10 は、この機械装置の初期状態やバルブ操作のタイミングなどの運転条件の記述に相当し、L6 および L8 は図 4 に示した状態遷移の表現記述である。

## 【0024】

HCCでは、運動方程式は、図から分かるようにプログラム内でそのまま記述することができる。また、それぞれの状態へ遷移する条件は、「always if」に続いて記述し、また、それぞれの状態から遷移していく条件は「watching」に続けて記述すればよい。

## 【0025】

なお、HCCでは、プログラムの記述の順序（図5における論理行番号L1→L10の順序）にそって実行されるわけではない。HCCでは、個別のプログラム記述のうち、シミュレーションを実行する時間軸に沿って成立するものが探索され、実行される。すなわち、論理行番号L1→L10の順序は実行順序とは関係がない。たとえば、シミュレーションを開始した時点では、L3およびL10のみが有効である。ここで、イベントRight (ev1) がL3により発生するため、L8の前提条件であるRightが有効となり、L8に記述されている運動方程式eq2が有効になる。つまり、図4の左側の状態からシミュレーションが実行されることになる。

## 【0026】

さらに、時間が50になるとL4が有効となり、イベントLeft (ev2) が発生し、L8の遷移条件（「watching」以下、すなわちLeft）が有効となって、L8の運動方程式eq2が無効となる。これに代わって、L6の前提条件が有効となり、運動方程式eq1が有効となる。

## 【0027】

なお、以上のようなプログラム例は外部からのイベント (ev3, ev4) によって状態が遷移する場合を記述したものであったが、勿論、内部の状況によって状態を変化させてもよい。たとえば、図2においてバルブ301が切り替えられない場合には、移動するピストン302がバネ303に接触し、該バネ303からの反力を受けるようになる。すなわちピストン302の位置に関して、外部からのイベントが無い場合でも状態遷移が起こる場合が存在する。このような場合は、例えばxが正であるかどうかの評価式（不等式）を評価することで状態遷移の必要性を判断できる。

## 【0028】

一般的に、ハイブリッドモデルは、常微分方程式や代数方程式を連立させた連立方程式によって表現される連続系モデルと、イベント発生に伴う状態遷移を表現するための状態遷移モデルとを組み合わせたモデルである。ハイブリッドモデルによれば、連続系モデルで表現される状態が外部からのイベントなどにより瞬時に切り替わるシステムを表現することができる。

## 【0029】

そして図1のモデル方程式制御情報解析部111は、以上説明した図5に示されるようなハイブリッドモデル記述104を入力とし、シミュレーション実行に先立ってモデル記述をまず解析し、必要となる連続系方程式のすべてをシミュレーションに適した形で登録する。

## 【0030】

具体的には、モデル方程式制御情報解析部111は、図5に示される運動方程式eq1や運動方程式eq2などの連続系方程式の記述と、同図に示されるRightやLeftなどのイベントに伴う状態遷移の制御情報の記述とをハイブリッドモデル記述104からシミュレーションの実行前に予め全て抽出するとともに両者を分離し、それぞれ、方程式構文解析部112およびモデル方程式制御情報記憶部113に渡す。

## 【0031】

モデル方程式制御情報解析部111は、イベントに伴う状態遷移の制御情報の記述に基づき、条件式とその条件式が成り立つ場合に成立する連続系方程式のID、およびこれに応じて無効になる連続系方程式のIDとを関連付けるテーブルを生成する。このテーブルにより表される情報がモデル方程式制御情報113である。連続系方程式切り替え部115は、シミュレーション実行過程において定期的にこのテーブルを参照し、その内容を順番にすべてチェックする。ここで、条件式が成立した場合には該当する連続系方程式IDについて、該連続系方程式を有効化又は無効化するための他のAPI (Application Interface) 関数を呼び出す。このAPI関数は、該連続系方程式について設けられた有効・無効を表すフラグをセット／リセットする。これにより連続系方程式の切り替えが実行さ

れる。なお、後述する第2実施形態では、上記のようなテーブル（モデル方程式制御情報113）を生成することに代えて、上記API関数を呼び出す第2のプログラミング言語（例えばC言語）のソースプログラムを生成する。

#### 【0032】

方程式構文解析部112は、与えられた連続系方程式を構文解析し、シミュレーション実行可能なデータ構造に変換して、方程式データ記憶部114に登録する。具体的には、方程式構文解析部112は個々の連続系方程式の記述を構文解析して複数の文字列に分解する。これら連続系方程式の記述を構成する文字列は、連続系方程式を登録するための所定のAPI（Application Interface）関数への引数として渡される。この方程式登録用API関数は、上記構文解析結果に基づき、連続系方程式の記述をシミュレーション実行可能なデータ構造（内部データ表現）に変換して、方程式データ記憶部114に登録する。なお、個々の連続系方程式には、ユニークなID番号が割り当てられる。

#### 【0033】

たとえば「 $ab/\cos(a-(c+b))-3c$ 」という式が与えられたと仮定すると、上記内部データ表現として図6のような木構造を生成する。この木構造において、例えば参照数字61は線形多項式の親ノード（節）、62は掛け算のノード、63は割り算のノード、64は外部関数（四則演算以外の意）のノード、65は線形多項式を構成する各項のノードを表している。本例において、木構造の葉に相当するものはすべて変数（ $a, b, c$ ）であり、これらに実数の係数が加わって線形式となる。線形式は $\cos$ などの外部関数の引数になったり、掛け算や割り算の対象となる。変数には、別途、値が確定しているかどうかのフラグが設けられており、またこのような木構造のデータに基づいて該変数の現在の値が保持される。木構造のすべての葉の値（すなわち変数の値）が確定していれば、式の値を計算することができる。方程式データ記憶部114では、式の値の計算などを高速に行うことができるように、予め内部のデータ構造をつなぎ合わせて木構造を構成してある。このように、方程式データ（木構造を有する内部データ表現）を予め準備しておく本実施形態によれば、例えば「 $ab/\cos(a-(c+b))-3c$ 」といった式の記述をシミュレーション実行時に構文解析する場合に比べ、シミュレーションを高速



に実行できるようになる。

#### 【0034】

ここで、本実施形態との比較例として、ハイブリッドモデルのプログラム記述をインタプリタ型の言語処理系にて実行させる場合を説明する。かかる比較例はたとえば従来技術で説明した米国ゼロックス社（商標）のHCCである。

#### 【0035】

インタプリタ型の場合、シミュレーション実行途中に方程式を追加することなどが許されるため、時間ステップを進めて数値積分を行うたびに、連立させるべき有効な方程式を列挙する処理、列挙された方程式を構文解析して内部の計算に適した内部データ構造（図6がその一例である）に変換する処理、方程式を連立させて実際の解く処理などを行わなければならない。これらの処理の際に、例えばどれだけの数の方程式が連立させられるかについても、あらかじめ決めることができないため、方程式が1つずつ列挙されるごとに必要なメモリを動的に確保する。処理によって必要なメモリの大きさは異なるため、あらかじめまとまったメモリ領域が確保されており、そこから必要なだけのメモリを確保することがよく行われる。また、その方程式が無効になった際にメモリを解放する処理を行わなければならない。開放されたメモリは後の処理で別の目的に使うことが可能であるが、小さな単位で細かく動的に確保されメモリは、解放された後にそのまま利用することが難しいから、ガーベジコレクション（ごみ集め；Garbage Collection）の処理によって、開放された細かなメモリの断片を集め、まとまった再利用可能なメモリ領域として再構成する必要がある。このような構文解析やメモリ管理の処理は計算コストが極めて高い複雑な処理であり、本来の目的であるシミュレーションの実行に影響を及ぼし得る。たとえば、ガーベジコレクションの処理が起動されてシミュレーションの実行が一時的に中断される場合などである。

#### 【0036】

これに対し本発明の実施形態では、上述したようにモデル方程式制御情報解析部111および方程式構文解析部112が設けられており、あらかじめハイブリッドモデル記述104をすべて解析し必要な内部データ表現（アクセスが高速で

かつ単純な配列などのデータ) を生成、管理する構成であるから、上記比較例のような問題は生じない。

#### 【0037】

シミュレーション実行時においては、ハイブリッドモデルシミュレーション実行部102が起動され、機構制御ソフトウェアシミュレータ108などから得られる制御信号106を受信しながら、連続系方程式の値を計算することでシミュレーション実行が行われる。このとき、連続系方程式切り替え部115はモデル方程式制御情報記憶部113の内容を参照し、連続系方程式の切り替えを有効・無効のフラグを用いて実行する。図2の状態では、図5の運動方程式eq1は無効であり、運動方程式eq2が有効になっている。ここで、Leftのイベントが発生した図3の状況においては、図5の運動方程式eq1を有効にし、運動方程式eq2を無効にするようフラグを操作する。これら有効・無効のフラグは方程式データ記憶部114に記憶される方程式それぞれの属性データとして管理される。

#### 【0038】

連続系シミュレーション部103は、方程式データ記憶部114を参照し、同記憶部114に木構造の形式で格納されている連続系方程式の内部データ表現を演算対象として、時間ステップづつ数値積分を実行する。シミュレーションは、常微分方程式及び代数多項式の連立からなる非線形連立方程式についての初期値問題である。このため、例えば図2に示される初期状態が与えられ、例えば一般によく使われているルンゲクッタアルゴリズムを用いて解の値を計算する。

#### 【0039】

必要なデータは機構シミュレータから出力を行い、さらに連続系方程式切り替え部115の処理に戻り、上記の処理を繰り返すことにより必要な時間のシミュレーションを実行する。シミュレーション結果は、変数値時間時歴記憶部105に保存され、シミュレーション終了後の分析などに利用される。

#### 【0040】

図7は、以上のように図1を参照して説明した機構シミュレータと、機構制御ソフトウェアとの連携を示す構成図である。図7において、機構シミュレータ（

ハイブリッドシミュレーション部 101) はブラックボックスとして簡易に示してある。

#### 【0041】

機構制御ソフトウェア (ないしは機構制御ソフトウェアシミュレータ) 108 は、シミュレーションの対象である機構の構成要素である例えばモータや、ソレノイドなどのアクチュエータへの動作コマンドを制御信号 106 として与える。これに応じてハイブリッドモデルシミュレーション部 101 はシミュレーション動作し、対象とする機構に設けられたフォトセンサーなどのセンサー情報を中心に必要なデータを生成して機構制御ソフトウェア 108 に送る。このような連携動作により、機構制御ソフトウェア 108 は、機構シミュレータ 107 との間で実機の機構と授受するものと同じデータを適正なタイミングで擬似的にやりとりすることができ、実機が存在しない状況におけるソフトウェアの検証作業などを実現できるようになる。

#### 【0042】

図 8 は、図 7 の構成に対しさらにキネマティクスシミュレーション部 109 を加えた場合を説明している。この図 8 の構成によれば、機構シミュレータ (ハイブリッドモデルシミュレーション部 101) は、キネマティクスシミュレーション部 109 に基づいて、例えばモータの軸の角度が時間的にどのように変化するかといった運動理論的解析情報をさらに求めることができる。各時間ステップにおいて、キネマティクスシミュレーション部 109 は、求めたモータの角度などから機構の姿勢などを計算することもできる。これにより、機構シミュレータ側では、例えばロボットのアームがフォトセンサーの光ビームを遮っていないかどうかなどの判定を行うことができる。つまり、3次元空間において複雑形状を有する機構物体が運動するような場合に、センサーの状態などを効率的に計算することができる。なお、キネマティクスシミュレーション部 109 の具体的な実現方法やこれに必要な要素技術に関しては、本願発明と同一出願人による出願に係る特開 2000-137740 公報、特開 2001-282877 公報、特開 2002-215697 公報等の記載が参考になる。

#### 【0043】

また、市販の機構解析ソフトウェアや3次元CADの機構シミュレーション機能などを利用することも可能である。また、図8の構成に関し、ハイブリッドモデルのようなモデル記述言語で機構の挙動をモデル化する機能は提供されていないが、特開2001-222572公報に記載の技術も参考にすることができる。

#### 【0044】

図9は、以上説明した本発明の第1実施形態に係るシミュレーションにおける一連の処理手順を示すフローチャートである。尚、この図9のフローチャートは、図8に示した構成、すなわちキネマティクスシミュレーション部109を備える構成におけるものである。

#### 【0045】

まず、図1のモデル方程式制御情報解析部111および方程式構文解析部112はハイブリッドモデル記述104を入力して構文解析する(ステップ501)。そして、イベントの発生に伴う状態遷移の制御情報の記述に基づき、条件式とその条件式が成り立つ場合に成立する連続系方程式のID、およびこれに応じて無効になる連続系方程式のIDとを関連付けるテーブルを生成し、モデル方程式制御情報記憶部113に記憶させる。また、連続系方程式の記述の構文解析結果に基づき、同記述をシミュレーション実行可能なデータ構造に変換するとともに方程式データ記憶部114に登録する(ステップ502)。

#### 【0046】

以上でシミュレーション実行のための前処理を終え、ここからシミュレーション実行の段階に移る。まずは機構制御ソフトウェアシミュレータ108等から与えられる制御信号106の受信処理を行う(ステップ503)。ここで、必要であれば、受信した制御信号に基づく値を適宜、変数に代入する。次に、状態変化の必要有無をステップ504にて判定する。状態変化が必要な場合、連続系方程式切り替え部115は、該当する連続系方程式について、その有効・無効のフラグを操作することにより、必要に応じて連続系方程式の切り替えを行なう(ステップ505)。

#### 【0047】

次に、連続系シミュレーション部103は数値積分を実行する（ステップ506）。次に、ハイブリッドモデルシミュレーション部101とキネマティクスシミュレーション部109との間で通信処理を行う（ステップ507）。具体的には、ハイブリッドモデルシミュレーション部101とキネマティクスシミュレーション部109との間で、モータ角度情報やセンサー情報などのやりとりが行われる。なお、図7のようにキネマティクスシミュレーション部109を備えない構成の場合、このステップ507はスキップされる。

#### 【0048】

次に、機構制御ソフトウェアシミュレータ108に対し機構シミュレータ側からセンサー情報などが送信される（ステップ508）。さらに、ステップ509にて終了判断を行ったのち、所定の終了条件が成立するまで時間を1ステップ進め（ステップ510）、ステップ503以降の処理手順を繰り返すことによりシミュレーション実行が行われる。

#### 【0049】

ハイブリッドモデルシミュレーション部101が機構制御ソフトウェアシミュレータ108やキネマティクスシミュレーション部109と連携し、全体として効率的にシミュレーションが実行されるためには、専らハイブリッドモデル記述104の内容に応じた処理を行う504、505、および506の各ステップと、専らデータ送受信等を行う503、507、および508の各ステップとの間の実行順序が適切かつ効率的に制御されなければならない。ここで、既に述べたように従来では、ハイブリッドモデル記述言語の特性（どの行が最初に実行されるか分からない）に起因し、上記実行順序を適切かつ効率的に制御するようプログラミングするには、極めて高度なテクニックを要し、難易度の高い作業である。また、単なる変数の値を扱うのではなく、外部から得られた情報を変数に代入してからハイブリッドモデルの処理を行うなど、外部とのインターフェースに関わる特別な外部関数などを用意し、これらをモデルの作成者が記述しなければならない。

#### 【0050】

これに対し本発明の実施形態によれば、ハイブリッドモデルシミュレーション

実行部 102 が図 9 に示した一連の処理を実行するよう構成されていることから、上記のような問題を回避できる。すなわち、機構制御ソフトウェアシミュレータ 108 等の外部（プロセス）との間の通信に相当するソフトウェア部分は、ハイブリッドモデルとは独立して予め固定的に組み込まれている。このようなソフトウェアモジュールを俗に「（シミュレーション）エンジン」などと称し、本発明に従う一の実施形態として提供される。なお、外部から得られたデータをどの変数に代入するかは、ハイブリッドモデル記述 104 とは別個に入力しておくことになるが、ハイブリッドモデル記述 104 を作成する際には、外部から得られたデータが代入される変数名だけ理解しておけばよく、通信等の手順を何ら気にすることなくモデル作成（プログラミング）が可能になる。

#### 【0051】

また、上述したように本実施形態によれば、シミュレーション実行に先立ってハイブリッドモデル記述 104 をまず解析し、必要となる連続系方程式のすべてをシミュレーションに適した形で登録しておくとともに、特に第 1 実施形態ではイベントの成立等に基づく状態遷移に係る実行制御のためのテーブルを生成しておく。シミュレーション時には連続系方程式切り替え部 115 が該テーブルを参照する。ここで、状態遷移に伴う方程式の切り替え（入れ替え）を行う必要があると判断した際には、該当する連続系方程式の有効・無効のフラグを操作（セット／リセット）することで簡単にこれを実現できる。このような仕組みによれば、シミュレーション中にモデル記述の構文解析やガーベジコレクション（Garbage Collection（GC）：ごみ集め）処理などに起因するシミュレーション実行速度の低下を防ぐことが可能になる。このような第 1 実施形態の機構シミュレータは、内部的には上記のような本発明に固有の処理手順を実行するものであるが、外見的にはインタプリタライクである。例えば、上記テーブルや方程式データを記憶する記憶部 113、114 等は RAM 等の記憶装置を用いて構成される。

#### 【0052】

##### （第 2 実施形態）

次に本発明の第 2 実施形態を説明する。

#### 【0053】

図10は本発明の第2実施形態に係る機構シミュレータの概略構成を示すブロック図である。第2実施形態の構成は、モデル方程式制御情報解析部111がハイブリッドモデル前処理部201としてハイブリッドモデルシミュレーション実行部102から独立した構成となっている点で第1実施形態とは異なっている。第2実施形態においては、ハイブリッドモデル記述104がまずハイブリッドモデル前処理部201により処理され、モデル方程式登録プログラム202およびモデル方程式制御プログラム203が生成される。また、ハイブリッドモデルシミュレーション実行部102を構成するソフトウェアモジュールとして、モデル方程式の登録を行うための関数および連続系方程式を切り替えるための関数がA P I (Application Program Interface) 関数として提供される構成である。

#### 【0054】

モデル方程式登録プログラム202およびモデル方程式制御プログラム203は、該当する上記A P I 関数を呼び出す記述を、入力されたハイブリッドモデル記述104に沿って適切に組み合わせたプログラムであり、モデル方程式制御情報解析部111により生成される。この観点から考えると、ハイブリッドモデル前処理部201は、入力をハイブリッドモデル記述104とし、出力を例えばC言語のA P I 関数呼び出しの記述を含むCプログラム（ソース）とするような、一種のコンパイラと考えることもできる。このようなモデル方程式登録プログラム202とモデル方程式制御プログラム203は、さらにC言語などのコンパイラによりコンパイルされ、例えば実行時に動的にリンク可能なライブラリが生成される。ハイブリッドモデルシミュレーション実行部102は、シミュレーション実行にあたっては、生成された動的リンクライブラリがリンクされ、これにより入力ハイブリッドモデルを忠実に再現するシミュレーションプログラムが完成し、実行可能になる。その実行時には、まず方程式構文解析部112を起動するA P I 関数が呼ばれ、その後に連続系切り替えのA P I 関数群が実行されてシミュレーションは遂行される。

#### 【0055】

ここで、図1に示した構成の第1実施形態と、図10に示した構成の第2実施形態との相違点をより詳細に説明する。

## 【0056】

それぞれの実施形態において、ハイブリッドモデルシミュレーション部102のアプリケーションインターフェースを構成する具体的なソフトウェアモジュールの仕様などは様々考えられるが、ここでは説明の都合上、以下の3つのAPI関数が最低定義されているとする。なお、プログラミング言語はC言語とする。

## 【0057】

```
int XXX_AddEqnData(char *eqn, int *err)
```

```
int XXX_ActivateEqn(int eqnid)
```

```
int XXX_DeActivateEqn(int eqnid)
```

1つ目のAPI関数XXX\_AddEqnDataは、1つの連続系方程式を表す文字列のポインタを引数に指定する。XXX\_AddEqnDataは、この連続系方程式を構文解析し、上述した図9のようなデータ構造に変換し、かかる内部データ表現を方程式データ記憶部114に登録する処理を行う。この処理において何らかのエラーが発生した場合には、errにエラーコードがセットされる。正常に処理が終了した場合は、登録された方程式のID番号を返り値とする。

## 【0058】

2番目のAPI関数XXX\_ActivateEqnは、引数に指定された方程式ID番号に相当する方程式を有効にする。もし、すでに有効となっている方程式が指定されている場合には何もしない。返り値はエラーコードである。

## 【0059】

3番目のAPI関数XXX\_DeActivateEqnは、XXX\_ActivateEqnとは逆に、引数に指定された方程式ID番号に相当する方程式を無効にする。すでに無効となっている方程式が指定された場合には何もしない。

## 【0060】

第1の実施形態においては、上述したようにモデル方程式制御情報解析部111によりハイブリッドモデル記述104から個々の方程式が抽出され、各々の方程式は更に方程式構文解析部112により複数の文字列に分解される。該方程式を構成する文字列はXXX\_AddEqnDataに引数として渡される。これにより方程式データ記憶部114に該方程式の内部データ表現が登録、蓄積される。同時に、個



々の方程式には、ユニークなID番号が割り当てられ、モデル方程式制御情報解析部111は、モデル制御情報記憶部113の内部データとして、条件式とその条件式が成り立つ場合に成立する方程式のIDおよび無効になる方程式IDとを表すテーブルを生成する。連続系方程式切り替え部115は、シミュレーション実行過程において定期的にこのテーブルを参照し、その内容を順番にすべてチェックする。ここで、条件式が成立した場合には該当する連続系方程式IDについて、該連続系方程式を有効化又は無効化するため、XXX\_ActivateEqnないしはXXX\_DeActivateEqnを呼び出す。

#### 【0061】

これに対し第2実施形態においては、モデル方程式制御情報解析部111がXX<sub>X</sub>\_AddEqnDataを必要な方程式について順に呼ぶ関数 (InitEqnData) を生成する。これがモデル方程式登録プログラム202に相当する。

#### 【0062】

また、モデル方程式制御情報解析部111は、シミュレーション実行の際に、時間が $\Delta t$ 進むごとに条件のチェックおよび方程式の変更（入れ替え）を行う関数 (ChangeEqn) も生成する。これはモデル方程式制御プログラム203に相当する。第1実施形態では、これが内部のテーブルデータとそれを参照する方程式切り替えメカニズムであったのに対し、第2実施形態では上記のように具体的なソースプログラムの形態で実現されている。例えば、図5に示したハイブリッドモデル例について、以下のようなC言語のソースプログラムが自動生成される。

#### 【0063】

```
static char eqn1[] = "f = m x' ' " ;
static char eqn2[] = "-f = m x' ' " ;
static int eqn1id;
static int eqn2id;
int InitEqnData()
{
    int err;
    eqn1id = XXX_AddEqnData(eqn1,&err);
```

```
    if(err != 0 ) return err;
    eqn2id = XXX_AddEqnData(eqn2,&err);
    if(err != 0 ) return err;
    return 0;
}

int ChangeEqn()
{
    int err;
    BOOL GetEvent(char *eventname);
    if( GetEvent(Left) ){
        err = XXX_ActivateEqn(eqn1id);
        if( err != 0 ) return err;
        XXX_DeActivateEqn(eqn2id);
        if( err != 0 ) return err;
    }
    if( GetEvent(Right) ){
        XXX_ActivateEqn(eqn2id);
        if( err != 0 ) return err;
        XXX_DeActivateEqn(eqn1id);
        if( err != 0 ) return err;
    }
    return 0;
}
```

なお、GetEventは、名前で指定されたイベントが生起しているかどうかをチェックする関数である。以上のプログラムは、上述したようにC言語のコンパイラによってコンパイルされ、さらに動的リンクライブラリの形式に整えられ、実行時にリンクされる。このように（ソース）プログラムが自動生成される本実施形態の場合、同ソースプログラム中に、eqn1idとeqn2idの宣言がなされており、2つの方程式しか出現していないことが明記されているから、実行時に割り当てるメ

メモリ量を、これら2つの方程式分について十分に用意すればよいことがわかる点で好ましい。なお、第1実施形態では、シミュレーション開始後にモデル制御情報記憶部113の内部データであるテーブルを原則的には拡張しない。シミュレーション開始前に最初にテーブルを生成する際には、モデル方程式制御情報解析部111から得られたテーブルの大きさに従って、該テーブルを保持するのに必要なメモリを確保すればよい。

#### 【0064】

なお、本発明は上述した実施形態に限定されず種々変形して実施可能である。

#### 【0065】

##### 【発明の効果】

以上説明したように、本発明によれば、ハイブリッドモデルを用いて複雑な機構系を簡便かつ正確にモデル化でき、該機構系を制御する制御ソフトウェアとの連携シミュレーションにも好適なシミュレーション方法およびプログラムを提供できる。

##### 【図面の簡単な説明】

【図1】 本発明の第1実施形態に係る機構シミュレータの概略構成を示すブロック図

【図2】 ハイブリッドモデル記述を説明するための具体例に係るシリンダ装置のある状態を示す図

【図3】 ハイブリッドモデル記述を説明するための具体例に係るシリンダ装置の別の状態を示す図

【図4】 ハイブリッドモデル記述を説明するための具体例に係るシリンダ装置の状態遷移を示す図

【図5】 ハイブリッドモデル記述の内容を示す図

【図6】 1つの連続系方程式を構文解析した結果得られる内部データ構造の説明図

【図7】 機構制御ソフトウェアと機構シミュレータの連携を示す構成図

【図8】 キネマティクスシミュレーション部を含む場合の構成図

【図9】 機構シミュレーションの処理手順を示すフローチャート

【図10】本発明の第2実施形態に係る機構シミュレータの概略構成を示すブロック図

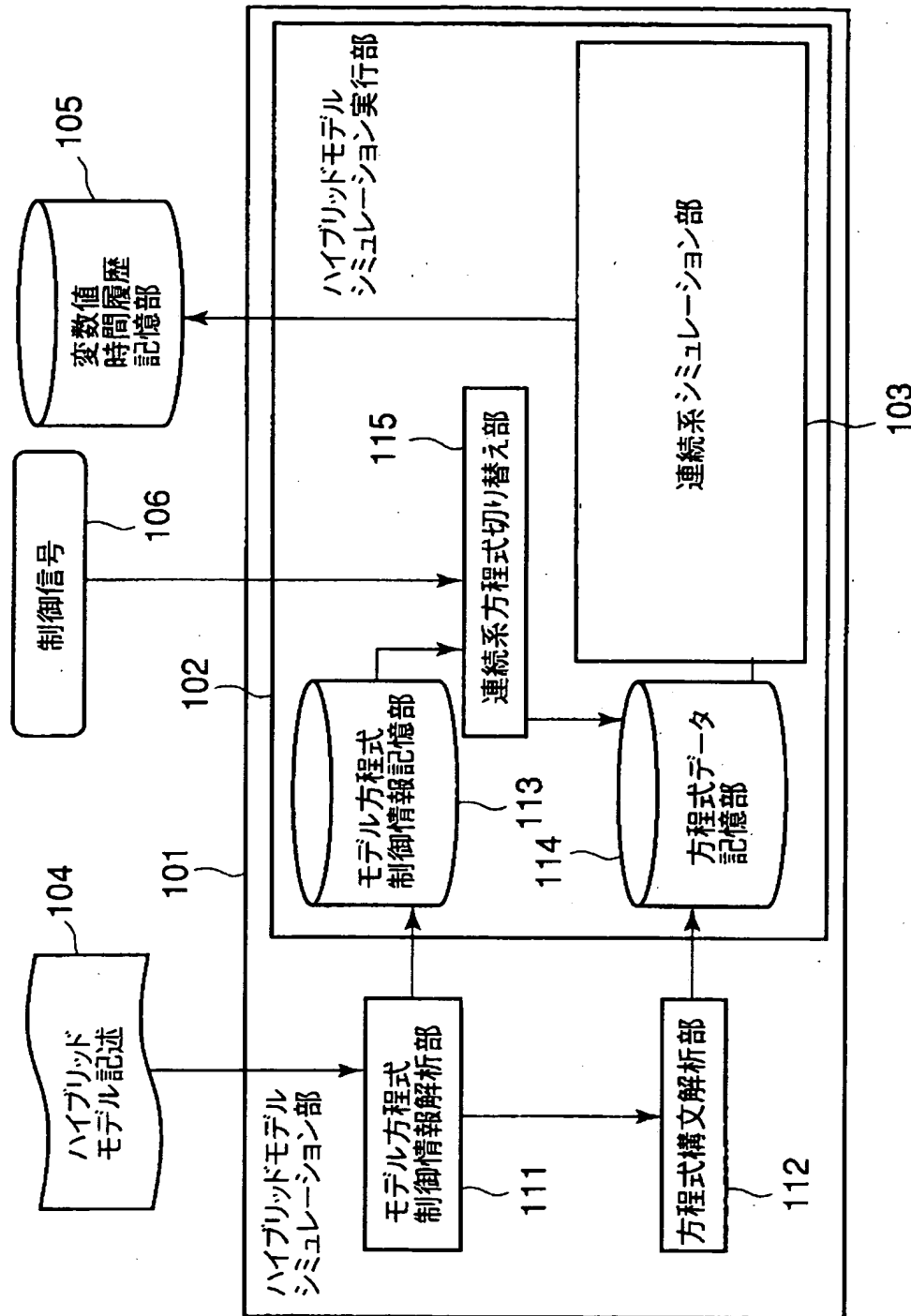
【符号の説明】

- 101…機構シミュレータ（ハイブリッドシミュレーション部）
- 102…ハイブリッドシミュレーション実行部
- 103…連続系シミュレーション部
- 104…ハイブリッドモデル記述
- 105…変数値時間履歴記憶部
- 106…制御信号
- 111…モデル方程式制御情報解析部
- 112…方程式構文解析部
- 113…モデル方程式制御情報記憶部
- 114…方程式データ記憶部
- 115…連続系方程式切り替え部

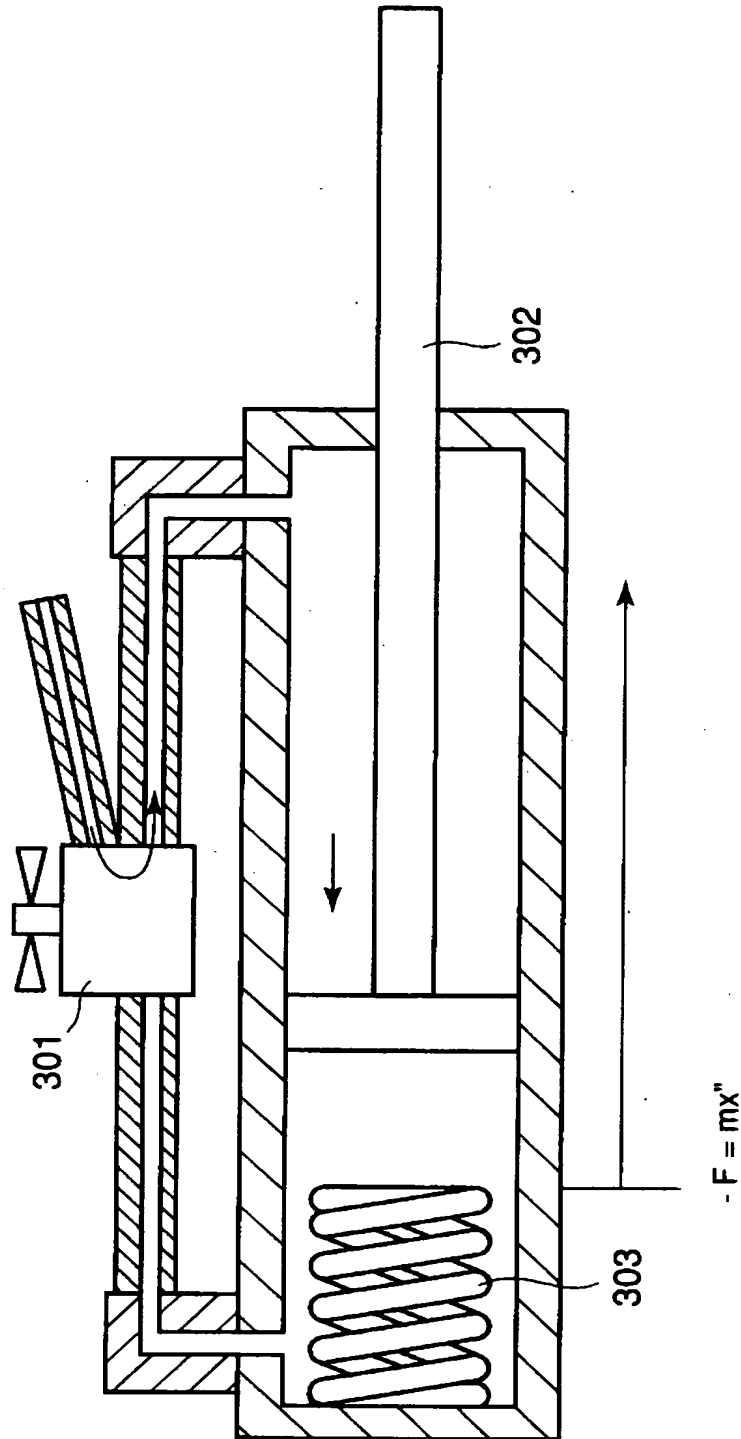
【書類名】

図面

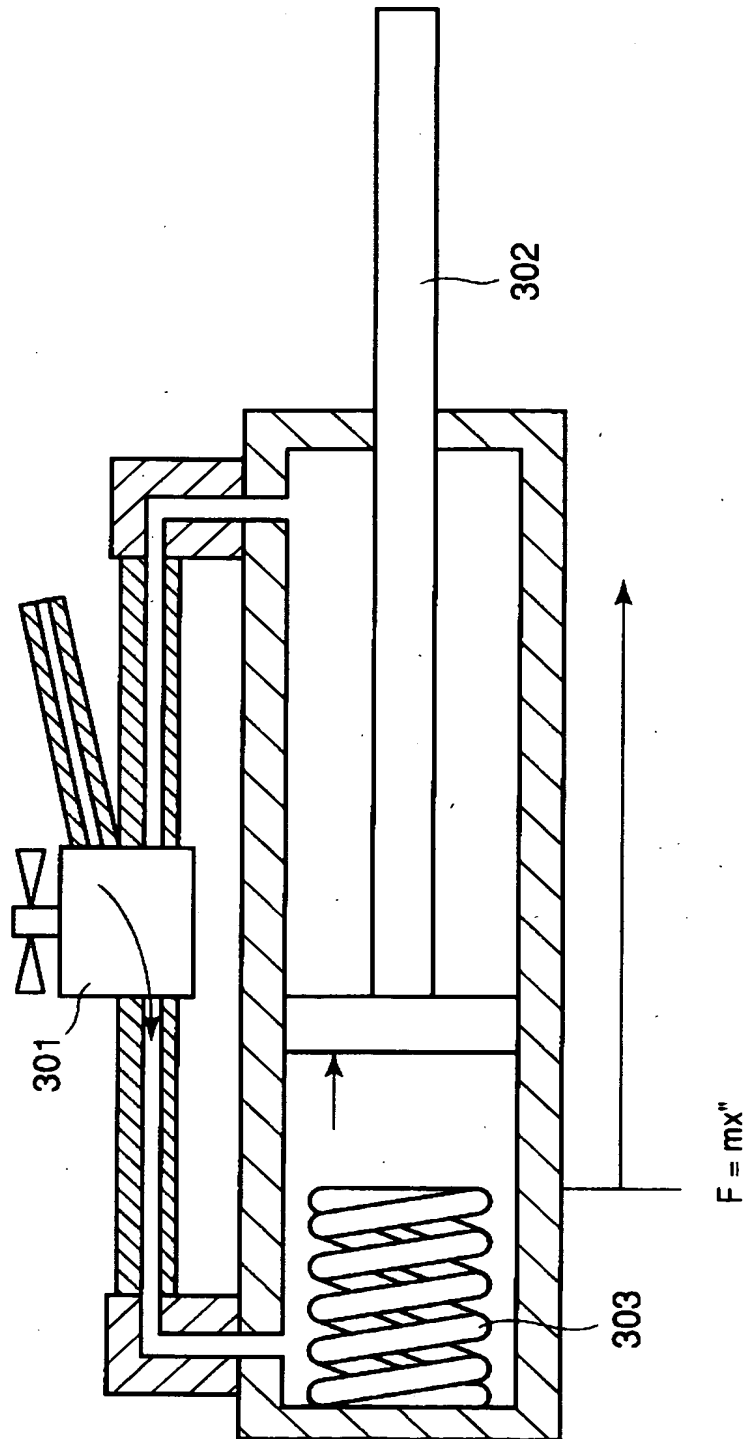
【図 1】



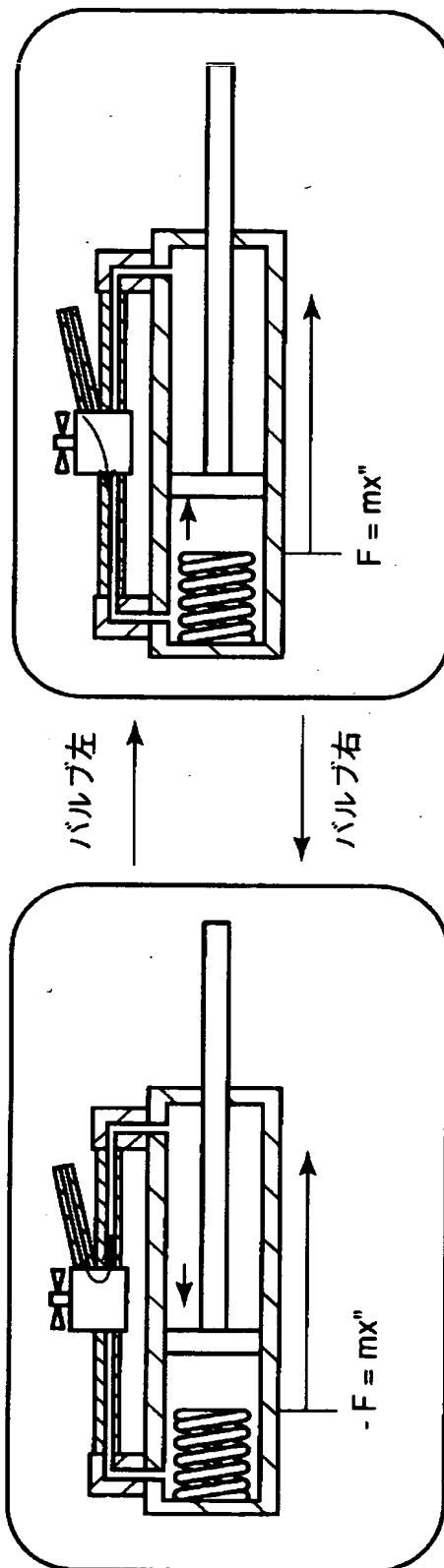
【図 2】



【図 3】



【図 4】

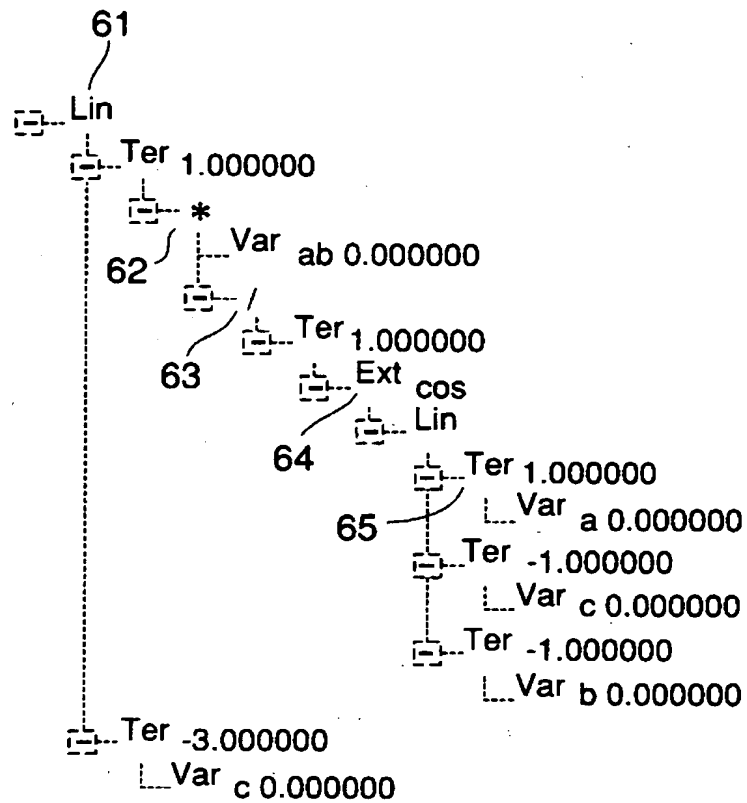




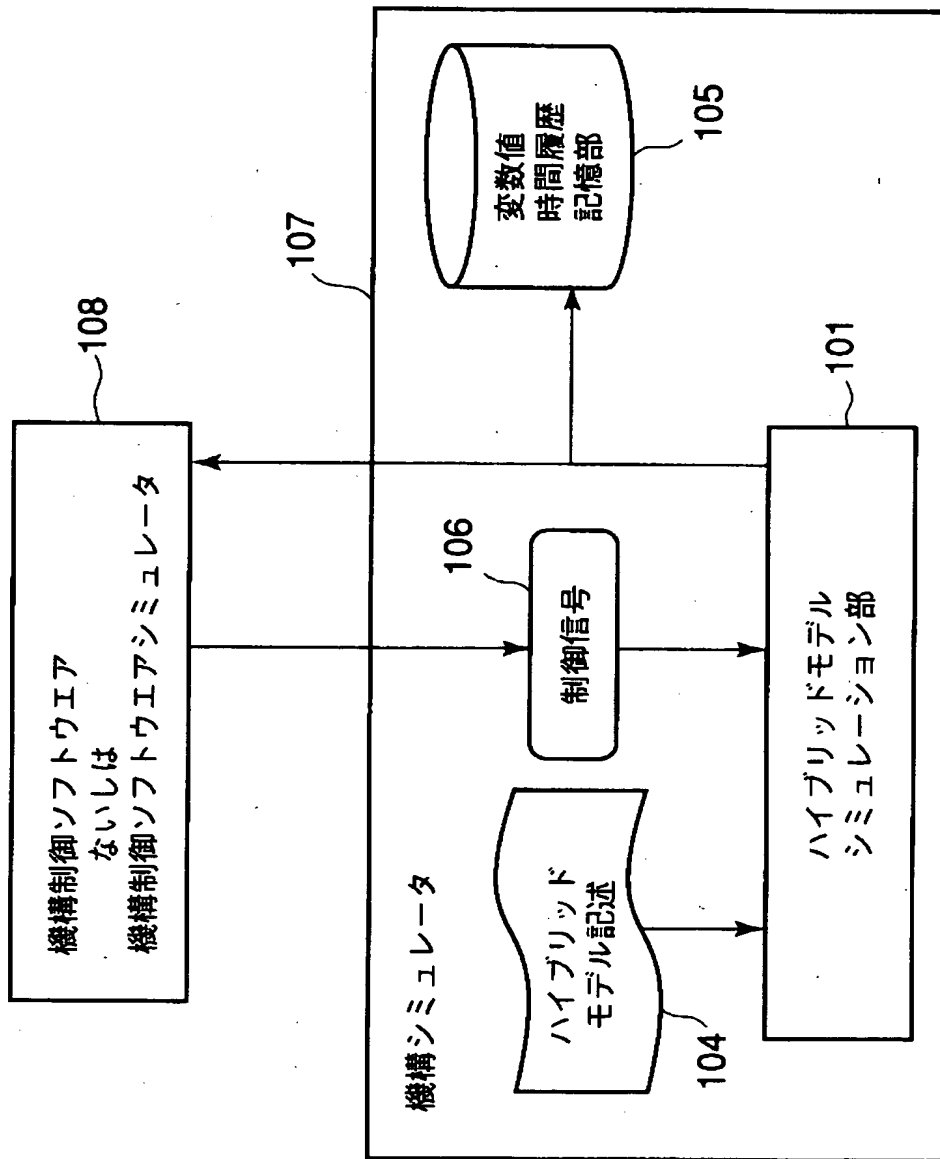
【図 5】

L1	# define m 1	
L2	# define f 100	
L3	Right,	ev1
		// バルブの初期状態
L4	wait 50 do Left,	ev2
		// 時間が50になるとバルブを右向きにする。
L5	// バルブが右向きの時の条件式	eq1
L6	always if Left then do always f = m * x " watching Right,	ev3
L7	// バルブが左向きの時の条件式	
L8	always if Right then do always -f = m * x " watching Left,	eq2
L9	sample (X),	ev4
L10	x = 0, x' = 0 // 変数xの初期状態	

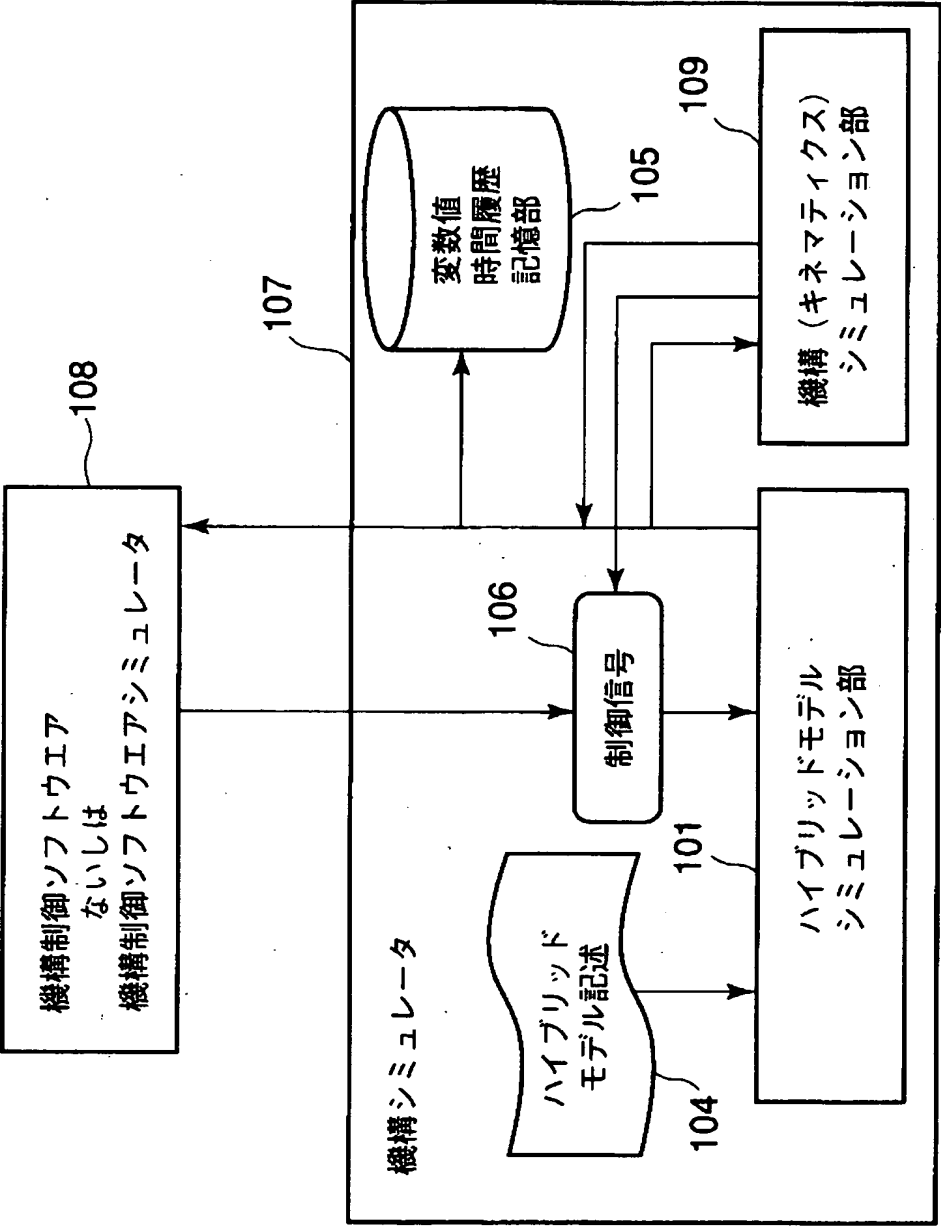
【図 6】



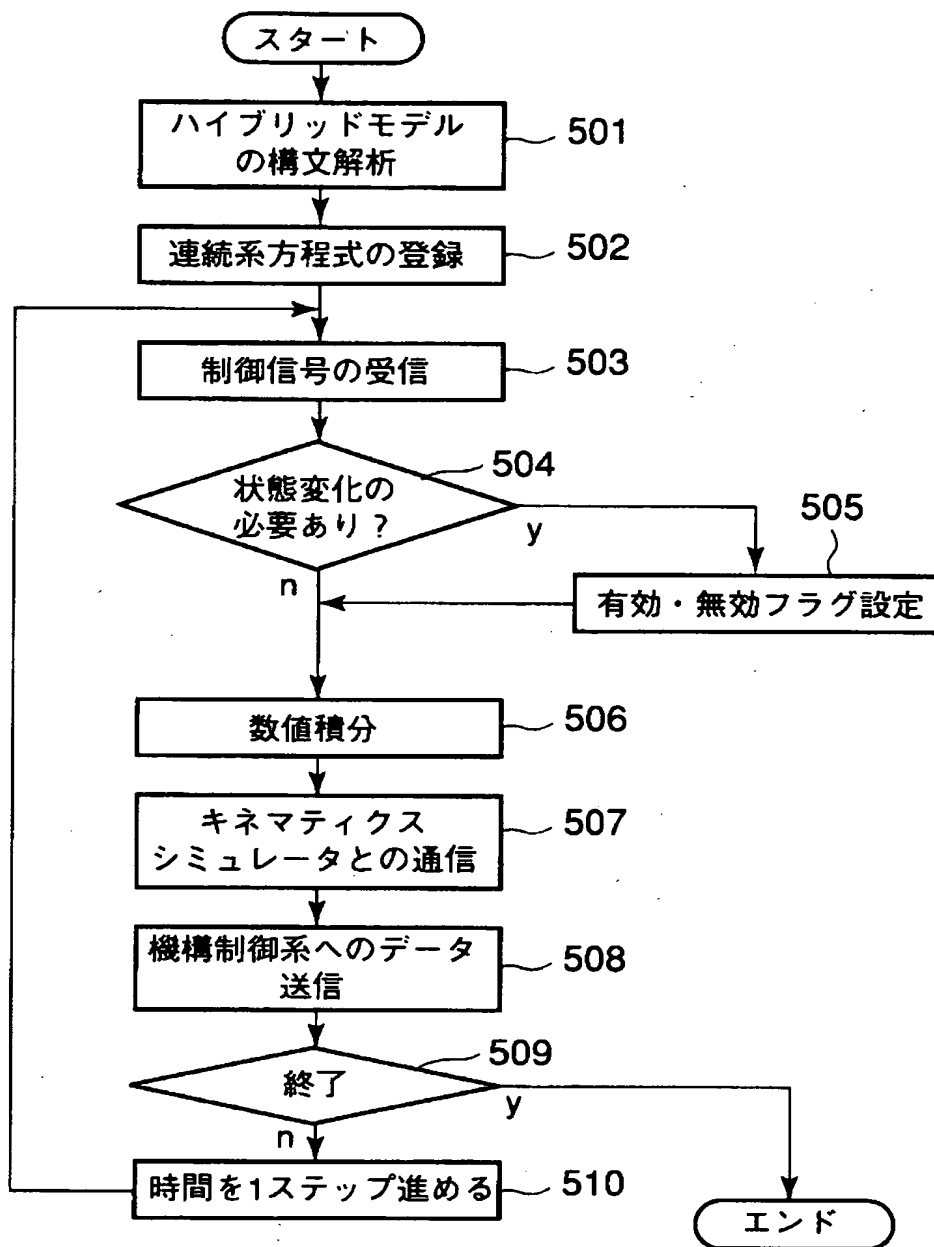
【図 7】



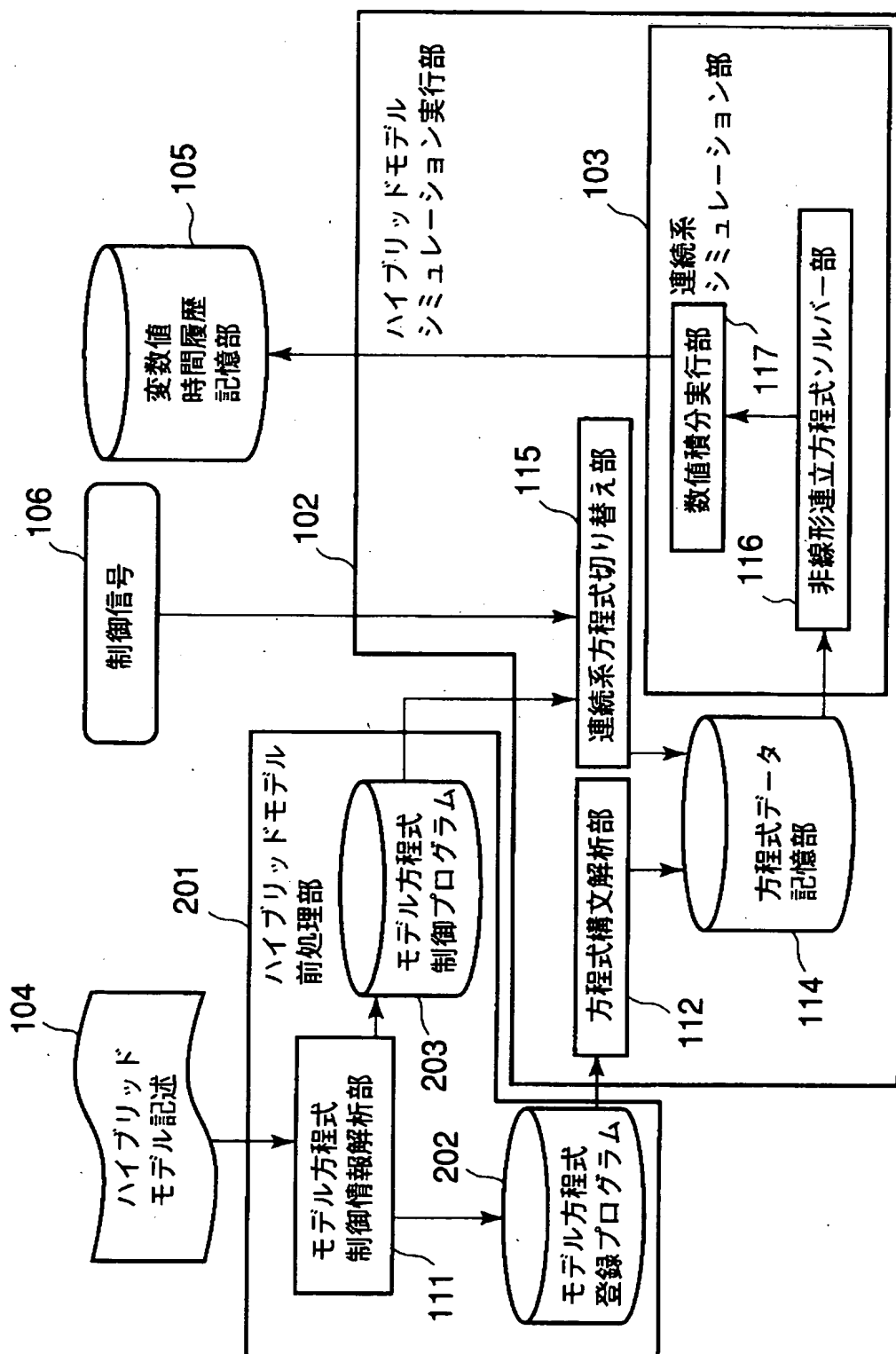
【図 8】



【図 9】



【図10】



【書類名】 要約書

【要約】

【課題】 ハイブリッドモデルを利用して対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーション方法およびプログラムを提供する。

【解決手段】 ハイブリッドモデル記述データを解析し、状態遷移に伴う連続系方程式の切り替えに関する記述と該記述データ中に出現する連続系方程式の記述とを抽出する。この結果から連続系方程式の切り替えが必要となる条件及び対象となる連続系方程式の関係を表すテーブルを生成する。また連続系方程式についての内部データ表現を生成する。イベントの生起に応じて上記テーブルを参照し、条件に従って連続系方程式を有効なものに切り替える。この切り替えられた連続系方程式に対応する内部データ表現を用い、時間軸に沿った数値積分により該連続系方程式を解いて上記機構の挙動を表すデータを出力する。

【選択図】 図 1

特願 2002-344228

出願人履歴情報

識別番号

[000003078]

1. 変更年月日      2001年   7月   2日  
   [変更理由]      住所変更  
                     住 所      東京都港区芝浦一丁目1番1号  
                     氏 名      株式会社東芝
  
2. 変更年月日      2003年   5月   9日  
   [変更理由]      名称変更  
                     住所変更  
                     住 所      東京都港区芝浦一丁目1番1号  
                     氏 名      株式会社東芝